Active Learning with Crowd-Sourcing

Semester Project Report

Alireza Ghasemi

Artificial Intelliegence Laboratory
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne

1 Introduction

There has been an explosive growth in the amount of available digital content during recent years. This is mostly because of the technological development and evolution of web which causes vast amount of digital media to be created. Managing and organizing such a large amount of data is a tedious task which has to be automated as much as possible. To facilitate this task, machine learning algorithms have been utilized for automatic classification of digital objects. However, achieving acceptable classification accuracy by machine learning algorithms requires large amount of labelled data to be used for training the algorithms. Labelling data has to be done manually and therefore it is a tedious time consuming and expensive task by itself. Therefore, many methods have been sought to make best use of limited resources in labelling training data for machine learning algorithms.

Active learning and crowd-sourcing are two complementary techniques for optimizing the cost of labelled data acquisition[2]. Active learning tries to improve classification accuracy by posing a limited number of queries to a user who can perfectly predict label of unlabelled data. It works by selecting among a large set of unlabeled data, the most informative data sample [3]. The informativeness of a sample is the expected amount of accuracy gain achieved after adding it to the training set. Many paradigms have been proposed to asses informativeness of data samples for active learning. One of the popular approaches is selecting the most uncertain data sample, i.e the data sample in which current classifier is least confident. Some other approaches are selecting the sample which yields a model with minimum risk or the data sample which yields fastest convergence in gradient based methods[8].

Another paradigm to reduce cost of data labeling is crowd-sourcing. Crowd-sourcing in its broadest form, "is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call"[7, 4]. In the case of data labelling, it means that instead of having a single expert who annotates data perfectly and precisely on demand, we deliver the task of labeling to a set of non-expert, usually anonymous crowd of people whose annotations are usually noisy and incomplete, but costs much lower than hiring an expert for perfect annotation. After collecting noisy annotations from the crowd, these annotations

should be aggregated in order to extract the most precise labellings from noisy annotations.

Crowd-sourcing and active learning are naturally complements of each other. Since the assumption of perfect error-free expert in traditional active learning is usually unrealistic and sometimes too expensive to obtain, we should expect noise and error in provided labels and deal with these noises to reduce their effects. Therefore, crowd-sourcing techniques for label aggregation can be used in active learning to efficiently reduce effects of noisy labels.

On the other hand, due to the limited annotation budget which is usually available in crowd-sourcing, this is highly desirable that the annotation budget is devoted to most informative unlabeled data so that we achieve best quality training set by spending as lower budget as possible. Therefore , it is reasonable to combine active learning with crowd-sourcing so that only those data which are most useful in classification are delivered to the crowd.

The combination of active learning and crowd-sourcing has been recently studied in the literature. [1] uses crowd-sourced active learning to train a machine translation system. It performs active learning by selecting for crowd-sourced translation those sentences which contain least overlap with already translated sentences. The work in [2] is another approach to combining active learning and crowd-sourcing with the goal of sentiment detection. It performs active learning by selecting the most diverse set of document for crowd-sourced sentiment classification such that a broad range of feature space is covered by labelled data. [12] and [6] are other approaches that combine active learning with crowd-sourcing.

The problem one usually faces when combining active learning with crowd-sourcing is that most crowd-sourcing systems give the users (crowd) freedom to select the task they want to work on. The reason is that users will make their best effort when they have the freedom to select the task they want to wok on. However, this is in contrast to the intuition of active learning to select the samples for which we need annotation. One way to overcome this problem is by packaging the annotation task as a game, which we will explain below.

1.1 Human Computation Games

A major problem in human computation and crowd-sourcing is the problem of motivation, i.e. how to motivate people to make their best effort in doing crowd-sourced tasks. It is quite important because budget is usually limited in crowd-sourcing and therefore it's significantly beneficial to motivate people so that best quality is achieved with minimal number of task repetitions.

Monetary motivation is a traditional approach for incentivizing workers which has been widely used and studied in the literature. Amazon's Mechanical Turk is a well-known example of a crowd-sourcing systems which is based on monetary rewards [6]. However, due to the natural limits on amount of budget one can devote to crowd-sourcing and also possibility of spamming and automatic answering, other more robust motivations methods have been sought.

Luis von Ahn, pioneer the field of human computation, proposed that crowd-sourcing tasks be packaged as interactive serious games so that people are moti-

vated to play a fun game using which a human computation task is actually done in each round of the game[10]. Using this method, entertainment becomes the motivating factor for people playing the game and therefore much larger-scale problems can be attacked.

The first human computation game was ESP, proposed by von Ahn [11] which aimed at semantically labeling a huge collection of images. The logic of the game was simple yet entertaining and useful for a wide range of applications. Two players are paired at random without knowing each other. At each game round, a single image is shown to both players and they are requested to type a word describing the game, as soon as the words of the two players match, they are both given a positive score whose values depends on various factors and the game continues with another image.

ESP was proven to be successful and about 1.3 millions of annotations were collected by 13000 players during a 4 month period[5]. Success of ESP, which was later acquired by Google and called Google Image Labeler, led to the introduction of many other games with various goals and different design. To name just a few, Verbosity is a game which aims at gathering common sense knowledge about words. KissKissBan is another game for image annotation. Peekaboom is used for image segmentation and "Phrase Detectives" is used to help constructing an anamorphic corpus for NLP tasks. [5] is a comprehensive survey of human computation games.

The main reason behind success of human computation games is the so-called "Gamification". Gamificiation is the process of packaging a problem into an interactive interesting game and construct solution of the problem using the data obtained from users by playing the game. It has been shown that good gamification can highly increase quality of the solutions found for many problem since it makes people pay more attention to the game (and problem) and therefore make better decisions.

In this work, we have designed and implemented a human computation game called "Suggest/Guess". Despite many traditional human computation games, Suggest/Guess has more than one principal goal. It aims and collecting a human-annotated dataset of sentiment labelled documents, and simultaneously construct a lexicon of highly polarized (positive and negative) words which can be used for sentiment detection tasks. We design the game to be as interesting as possible and also attract maximal attention of the players, so that we can collect high quality information from the game rounds.

2 Suggest/Guess Game

Suggest/Guess is a human computation game aimed at annotating a dataset of review texts based on their sentiment (whether they are positive or negative) and simultaneously obtaining a lexicon of strong positive and negative words for research purposes. Having two products as the result of playing instead of merely trying to obtain document annotations is the most discriminating factor of Suggest/Guess. The idea of using crowd-sourcing for feature extraction has

already been discussed in [9], but not as a human computation game. In the rest of the following section, we will discuss the game play and rules of Suggest/Guess.

2.1 Rules of Suggest/Guess

Suggest/Game is a two-player asynchronous game. It aims at annotating a large corpus of text documents like what ESP does for images. However, Suggest/Guess does this in a different way because of its rules and asynchronous approach. The differences allow Suggest/Guess to obtain more useful information from played game rounds than ESP does.

The two players of the game are called "Suggester" and "Guesser". These roles are initialized randomly and interchanged after each round of the game.

The Suggester, who starts each round will be given a full text document and he/she is supposed to:

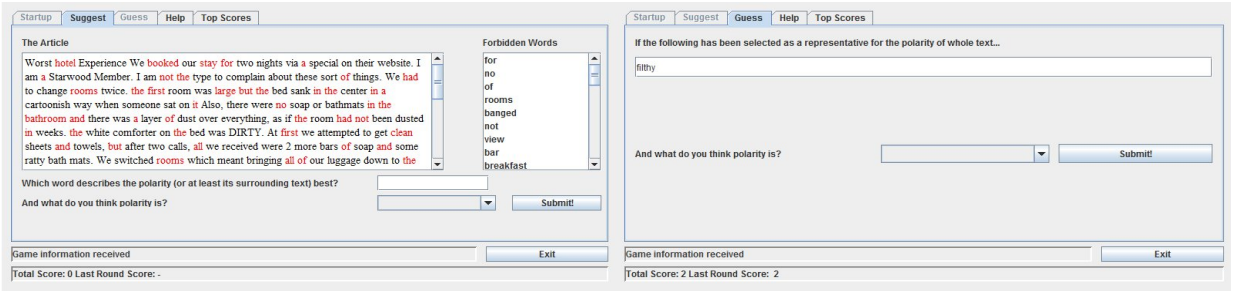
1. Decide whether the whole text is positive or negative, i.e. the author is praising about a subject or criticising it.
2. Select a single word (or a sequence of words, as short as possible) which best describes the polarity (positive or negative) he has selected in part (1). For example, when the negative polarity is chosen, the word "terrible" would be a good choice for the representative word (provided that it exists in the text).

The Guesser, on the other hand, will be given only the word (or word sequence) suggested by the Suggester (he won't see the whole text) and he has to guess polarity of the whole text just based on that single word. If the polarities suggested by the two players are the same, they are both given some positive score (based on factors described below) otherwise 0. Then the roles are interchanged and the game continues with a new document.

The guesser can also refuse to make a guess about polarity of the text (when for example the suggested word is ambiguous or not a good representative) in which case the suggester has two more opportunities to suggest another word from text.

Suggest/Guess is a cooperative game. It means that the two players are not opponent and they both receive equal score after each round (Not high score for one player and low score for the other). Therefore, the Suggester should make his best efforts to select the most polarized word from the test which best describes the selected sentiment or polarity. The UI screens for Suggester and Guesser are depicted in figures 1a and 1b respectively.

Scoring The score of each suggested word (or word sequence) depends on a variety of factors, including the length of the sequence and its novelty, i.e. how many times it has already been selected by other players. Suppose that the word sequence w is present in the current text document and also it has been present in text documents of n_w of already played game rounds. Assuming w has been selected k_w time before current game round, the potential score of w, PS_w is defined as:



(a) Suggester Role (b) Guesser Role

Fig. 1: The Suggest/Guess UI.

$$PS_w = \left[\frac{1}{length(w) \times \frac{k_w}{n_w}} \right] \tag{1}$$

In (1), $length(w)$ is the length (number of words) of sequence w. Using this scoring strategy, long word sequences are penalized and therefore players are encouraged to select as shortest sequences as possible. Single words that are not already selected by other players will yield highest score. In the game UI, score of each selected word will appear on the status bar while being typed.

Moreover, some words are not allowed to suggest and will yield zero score regardless of the agreement in polarity judgements. These words are coloured red in the text and are separately displayed in the forbidden list.

The cooperation between the Suggester and the Guesser and the requirement of agreement between them for achieving more scores, allows us to collect precise annotations and simultaneously build a good quality lexicon of words which are most important in detecting polarity of the text, either positive or negative.

The total score of each player is displayed on the scoreboard at the bottom of the Suggest/Guess graphical user interface.

3 Experiments

3.1 Implementation Details

The game was implemented as a traditional three-tier web application. For data storage, we used H2 embedded database which has proven to be fast enough for scientific information retrieval applications.

The server side of the application was implemented using Java and the Play! framework, which is a lightweight easy to use framework for Java MVC web application framework.

The client side of the game was implemented using Java Applet technology. We used a service oriented approach to define interactions between the client

and the server so that the game play is defined as a sequence of HTTP requests between client and server. Using this approach,, client and server are maximally separate and therefore various client applications can be written, e.g. to run on smart phones.

3.2 Experimentation Environment

A subset of 1000 articles from the Hotel Review dataset were chosen randomly to be used in game round. More than 20 players played the game during a period of one month. With their efforts, 697 annotations were collected during this period.

Moreover, a total of 318 distinct words and word sequences were selected by players as important positively or negatively polarized words. These words formed our lexicon for further studies and experiments.

For selecting articles for each round (i.e. active learning), a combination of strategies were used. From the set of documents which have not already been already labelled by any of the players, we select the article with the least difference between number of positive and negative (as collected in the lexicon constructed so far) words so that we get the most information from annotations. If all document have been annotated at least two times, we select among documents that for which the two annotations disagree, so that we solve disagreements by majority vote.

3.3 Quality of Annotations

The hotel reviews dataset contains ratings as well as review texts which have been provided by review authors themselves. These rating, give a score of 1 (most negative) to 5 (most positive) to the described hotel which is obviously correlated with the inherent polarity of the text.

We used review ratings as a ground truth to study quality of player annotations. We considered ratings higher than 3 as positive and lower than 3 as negative. Review with rating equal to 3 were considered neutral and were excluded from further experiments.

Comparing user-provided annotations with that of ratings showed promising results. We noticed a 90% match between these two sources. For comparison purposes, we also trained a binary classifier using bag of features to detect polarity of reviews. Results are summarized in table 1.

A direct comparison between gamification and traditional crowd-sourcing can be made by looking at the first two rows of table 1a. The first row shows the precision of the annotations provided by players of the game, whereas second row shows results of a previous survey collected from a group of 27 users who were asked to predict sentiments of the a subset of documents. We can see that merely packaging the question as a game significantly improves accuracy of the results.

We can infer from table 1 that gamification helps a lot in obtaining good quality annotation results. Therefore, annotations derived from players' effort are highly reliable and can be used for further studies.

Table 1: Comparison of Game Annotation Accuracies With that of Automatic Classifiers

Method	**Accuracy**
Game Collected Annotations	**90.4**
Aggregated Crowd Votes	82.5
Naive Bayes	80.5
Logistic Regression	83.6
SVM	82.8

3.4 Quality of the Collected Lexicon

Assessing quality of the Lexicon is not as direct as that of annotations. We studied quality of the collected lexicon in two ways. Here we mean by Lexicon the set of words and word sequences (n-grams) suggested by all players.

In the first method, we used the collected lexicon as the feature vector to extract features from review texts. We used simple binary features which denote absence or presence of each word (feature) of the lexicon in the document. Therefore, A binary feature vector of size 318 (total number of distinct lexicon words) were formed for each document which is quite shorter than 8800 features of the Bag of Words approach.

The second approach is even more compressed and more naive. This time we simply count the number of positive and negative words (as explained by game players by their suggestions) in each review text and take the difference as a single feature to decide about polarity of a word. This method is aggressively simple but extremely fast.

As well as storage efficiency, we see in table 2 that the classification accuracy also increases significantly when using game collected lexicon instead of the full dictionary of all words in the document set. In this table, accuracy of Naive Bayes classification which is a simple yet powerful and fast text classification method is tested with various kinds of feature vectors. We see that using only top 200 elements of the lexicon (in terms of frequency of use by players) beats other methods and achieves best accuracy.

Moreover, the last row of table 2 shows the result of using single word count difference feature. We can see that although discriminative accuracy of this method is not as high as other approaches, the extremely high training and testing speed and the fact that a simple thresholding approach can be used as classifier could make this method to be considered for real-time applications.

4 Conclusion and Future Works

In this report we introduced Suggest/Guess, a human computation game designed for simultaneous feature extraction and sentiment annotation. We conducted experiments to study how effective the gamification is and how precise

Table 2: Comparison of Naive Bayes Classification Accuracy Using Different Feature Vectors

Feature Extraction Method	**Accuracy**
Bag of Words	80.5
Full Lexicon	87.8
Top 100 Lexicon	82.7
Top 200 Lexicon	**88.2**
Word Count	83.7

the quality of annotations and extracted lexicon are. We showed that packaging the problem of sentiment classification as a game significantly improves the quality of obtained annotation.

The idea of the game could be further extended by testing other scoring functions to better motivate players and various document selection strategies to have a better trade-off between informativeness and interestingness. Moreover, a smart automatic player could be designed to perform active learning on feature extraction and direct the word suggestion process toward selecting more distinctive features.

References

1. Ambati, V., Vogel, S., and Carbonell, J. Active learning and crowd-sourcing for machine translation. *Language Resources and Evaluation (LREC)* (2010).
2. Brew, A., Greene, D., and Cunningham, P. Using crowdsourcing and active learning to track sentiment in online media. In *ECAI* (2010), vol. 215 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 145–150.
3. Ghasemi, A., Rabiee, H. R., Fadaee, M., Manzuri, M. T., and Rohban, M. H. Active learning from positive and unlabeled data. In *ICDM Workshops* (2011), IEEE, pp. 244–250.
4. Howe, J. The rise of crowdsourcing. *Wired magazine 14*, 14 (2006), 1–5.
5. Krause, M., and Smeddinck, J. Human computation games: A survey.
6. Laws, F., Scheible, C., and Schütze, H. Active learning with amazon mechanical turk. In *EMNLP* (2011), ACL, pp. 1546–1556.
7. Quinn, A., and Bederson, B. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 2011 annual conference on Human factors in computing systems* (2011), ACM, pp. 1403–1412.
8. Settles, B. Active learning literature survey. Tech. rep., 2010.
9. Settles, B. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. *Conference on Empirical Methods in Natural Language Processing (EMNLP), Edinburgh, Scotland, July* (2011).
10. Von Ahn, L. Games with a purpose. *Computer 39*, 6 (2006), 92–94.
11. Von Ahn, L., and Dabbish, L. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2004), ACM, pp. 319–326.
12. Yan, Y., Rosales, R., Fung, G., and Dy, J. G. Active learning from crowds. In *ICML* (2011), Omnipress, pp. 1161–1168.